数理解析研究所講究録 第2108巻 2019年 1-13

線形半無限計画問題に対する切除平面法の高速化

京都大学・大学院情報学研究科 引間泰成
東北大学・大学院情報科学研究科 林俊介
Yasunari Hikima*
Graduate School of Informatics, Kyoto University
Shunsuke Hayashi
Graduate School of Information Sciences, Tohoku University

概要

線形半無限計画問題 (LSIP) とは，有限次元の変数をもつ線形関数を，無限個の線形不等式で表される制約領域上で最小化する問題であり，フィルタ設計や汚染費用問題など多くの応用が知られている．LSIP に対する解法アルゴリズムとして切除平面法が知られているが，この手法では毎回の反復で生成される部分問題を厳密に解くことが前提となっている．そこで，本稿ではこの部分問題を非厳密に解くことによって切除平面法を高速化するアプローチを提案し，アルゴリズムの収束解析を行う．数値実験ではテスト問題に対して提案した高速化アプローチを適用し，特に高次元の問題に対して，提案アプローチの方が既存の切除平面法よりも高速に解が得られることを確認する．

1 序論

線形半無限計画問題 (Linear Semi-Infinite Programming: LSIP)[2, 3] とは，有限次元の変数空間において，無限個の線形不等式制約で表される制約条件の下，線形関数

* email: hikima.yasunari.62x[AT]st.kyoto-u.ac.jp

を最小化する問題[*1]であり，一般に次のように定式化される．

$$\begin{aligned} &\underset{x\in\mathbb{R}^n}{\text{Minimize}} && \boldsymbol{c}^\top \boldsymbol{x} \\ &\text{subject to} && \boldsymbol{a}(\boldsymbol{t})^\top \boldsymbol{x} - b(\boldsymbol{t}) \geq 0 \quad (\forall \boldsymbol{t} \in T). \end{aligned} \tag{1}$$

ここで $T \subseteq \mathbb{R}^s$ は空でないコンパクト集合であり，$\boldsymbol{c} \in \mathbb{R}^n$ は与えられた定数ベクトルを表す．また $\boldsymbol{a} : T \to \mathbb{R}^n$ および $b : T \to \mathbb{R}$ は与えられた連続関数である．ベクトル $\boldsymbol{t} \in \mathbb{R}^s$ は有限個の制約条件を含むような通常の最適化問題における制約関数の添字に相当するので，本稿では $\boldsymbol{t}$ を添字ベクトル (または単に添字)，T を添字集合と呼ぶことにする．なお，LSIP には有限個の等式制約を含めることができるが，表記の簡易性のため本稿では省略する．

典型的な LSIP の例として，チェビシェフ近似問題 [3] がある．この問題は，閉区間 $[\alpha, \beta]$ 上で定義される関数 $\varphi(t)$ を多項式 $P(t, \boldsymbol{x})$ で近似する問題である．近似の度合いを誤差の一様ノルム[*2]で測るとすれば，誤差の最大値を最小化する問題として次のように定式化することができる．

$$\underset{x}{\text{Minimize}} \max_{t\in[\alpha,\beta]} |P(t, \boldsymbol{x}) - \varphi(t)| \tag{2}$$

ここで，補助変数 $\eta > 0$ を導入することによって問題 (2) は次のような問題に書き換えることができる．

$$\begin{aligned} &\underset{(x,\eta)\in\mathbb{R}^{n+1}}{\text{Minimize}} && \eta \\ &\text{subject to} && -\eta \leq P(t, \boldsymbol{x}) - \varphi(t) \leq \eta \quad (\forall t \in [\alpha, \beta]). \end{aligned} \tag{3}$$

ここで，多項式 $P(t, \boldsymbol{x})$ が実係数の多項式 $P(t, \boldsymbol{x}) := \sum_{i=0}^{n-1} x_i t^i$ のように与えられるとき，問題 (3) の目的関数および制約関数はいずれも変数 $(\boldsymbol{x}, \eta)$ に関して線形である．また，変数 $(\boldsymbol{x}, \eta)$ の次元は有限で，制約条件の個数は無限であるので問題 (3) は LSIP に他ならない．

チェビシェフ近似問題 (3) の他にも，水資源管理問題 [6] やフィルタ設計問題 [10] など多くの応用が知られている．そのため，LSIP を効率的に解くためのアルゴリズムの研究がこれまで盛んに行われてきた．

LSIP に対する多くのアルゴリズムでは，無限個の制約条件を有限個に緩和した問題を繰り返し解くことが基本となる．無限個の制約条件を有限個に緩和するとは，LSIP(1) における集合 T を，有限個の要素からなる T の部分集合 $T' := \{\boldsymbol{t}_1, \boldsymbol{t}_2, \ldots, \boldsymbol{t}_m\}$ に置

[*1] 目的関数を最大化する問題は，目的関数値を (-1) 倍することで最小化問題へ帰着できる．

[*2] 集合 S 上で定義される実数値関数 $f(x)$ に対する一様ノルム $\|f\|_\infty$ は，$\|f\|_\infty := \sup_{x\in S} |f(x)|$ で定義される．

き換えることを意味する．この問題を LP(T') と表すと，LP(T') は次のような最適化問題として記述される．

$$\begin{array}{ll} \underset{x\in\mathbb{R}^n}{\text{Minimize}} & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{subject to} & \boldsymbol{a}(\boldsymbol{t}_i)^\top \boldsymbol{x} - b(\boldsymbol{t}_i) \geq 0 \quad (i = 1, 2, \ldots, m). \end{array} \tag{4}$$

問題 (4) の目的関数は $\boldsymbol{x}$ に関して線形であり，制約条件は有限個の線形不等式で表される．このような最適化問題は線形計画問題 (Linear Programming: LP) と呼ばれ，単体法や内点法などの既存のアルゴリズムによって効率的に解くことができる [12].

LSIP に対するアルゴリズムとしてよく知られている手法に，離散化法 (discretization method) [5, 8, 9] と交換法 (exchange method) [5, 11] がある．離散化法は，$|T_k|$[*3]$< \infty$ かつ $\lim_{k\to\infty} \mathrm{dist}(T_k, T) = 0$[*4]を満たすような T のグリッド部分集合の列 $\{T_k\} \subseteq T$ を逐次生成する．そして，$k \to \infty$ とすることによって，LP(T_k) の最適解 $\boldsymbol{x}^k$ を解くべき LSIP の最適解へ収束させることを考える．この手法は直感的にも分かりやすく，アルゴリズムの実装も比較的容易である. しかし，k が大きくなるにつれて部分問題 LP(T_k) の制約の個数が無限に大きくなるという問題点があり，必ずしも効率的な手法とは言えない．一方，交換法では，T_k に属する要素 $\boldsymbol{t}^k_{\mathrm{rmv}}$ と $T \setminus T_k$ に属する要素 $\boldsymbol{t}^k_{\mathrm{new}}$ に対して，$T_{k+1} := T_k \setminus \{\boldsymbol{t}^k_{\mathrm{rmv}}\} \cup \{\boldsymbol{t}^k_{\mathrm{new}}\}$ とすることによって集合 T_k を更新することを考える．この手法は，集合 T_k の要素数 $|T_k|$ は上に有界であるので，離散化法のように制約の個数が無限に多くなることは一般にはない．しかし，LP(T_k) の最適解 $\boldsymbol{x}^k$ を解くべき LSIP の最適解へ収束させるためには，交換要素である $\boldsymbol{t}^k_{\mathrm{rmv}}$ と $\boldsymbol{t}^k_{\mathrm{new}}$ を上手く選ぶ必要がある．

これらのアルゴリズムとは別に，切除平面法 (cutting plane method)[1, 11] と呼ばれる解法がある．この手法では $T_k \subseteq T$ かつ $|T_k| < \infty$ を満たすような T のグリッド部分集合の列 $\{T_k\}$ を，$T \setminus T_k$ に属する要素 $\boldsymbol{t}_{\min}$ を上手く選び，$T_{k+1} := T_k \cup \{\boldsymbol{t}_{\min}\}$ とすることで更新する．そして，$k \to \infty$ とすることによって LP(T_k) の最適解 $\boldsymbol{x}^k$ を LSIP(1) の最適解へ収束させることを考える．ここで，新たに加える添字 $\boldsymbol{t}_{\min}$ の典型的な選び方として，下位問題

$$\underset{t\in T}{\text{Minimize}} \left\{ \boldsymbol{a}(\boldsymbol{t})^\top \boldsymbol{x}^k - \boldsymbol{b}(\boldsymbol{t}) \right\} \tag{5}$$

の大域的最適解を求めることが知られている．このとき，切除平面法のアルゴリズムによって生成される点列の集積点は，LSIP(1) の最適解であることが証明されている [3].

しかしながら，この手法における問題点として次の 2 点が挙げられる．まず，(i) 毎回の反復で $\boldsymbol{t}$ に関する最適化問題 (5) を解かなければならない点である．特に，集合

[*3] 集合 E の要素数を $|E|$ で表す．

[*4] $X \subset Y$ であるような二つの集合間の距離を $\mathrm{dist}(X, Y) := \sup_{y\in Y} \inf_{x\in X} \|x - y\|$ で定義する．

T が高次元の場合や，T 上の関数 $\boldsymbol{a}(\boldsymbol{t})$, $b(\boldsymbol{t})$ が凸でない場合，問題 (5) の大域的最適解を求めることは一般に困難である．次に，(ii) 部分問題 LP(T_k) を厳密に解くことを前提としている点が挙げられる．LP(T_k) は線形計画問題であるため，非線形最適化問題に比べると高速に解くことが期待できる．しかし，決定変数の次元が大きい場合や制約条件の個数が多い場合，解を求めるためにより多くの計算コストがかかる．(i) の問題に関して，Wu, Fang, Lin[11] は問題 (5) の大域的最適解 $\boldsymbol{t}_{\min}$ を求める代わりに，十分小さな正の定数 $\delta > 0$ に対して，$\boldsymbol{a}(\boldsymbol{t})^\top \boldsymbol{x}^k - b(\boldsymbol{t}) < -\delta$ を満たすような添字 $t_{k+1} \in T \setminus T_k$ を求め，$T_{k+1} := T_k \cup \{\boldsymbol{t}_{k+1}\}$ と更新するアルゴリズム (Relaxed Cutting Plane Method: RCP 法) を提案した．Wu et al. はこの手法によって離散化法や既存の切除平面法よりも効率的に問題が解けることを数値実験で確認している．しかし，RCP 法では部分問題を厳密に解くことを前提としており，(ii) の問題は解消されていない．

2　準備

本節では，LSIP に対する解法アルゴリズムを記述するために必要となる前提知識について述べる．

2.1　線形計画問題

1 節で述べたように，LSIP に対する解法アルゴリズムは，部分問題である線形計画問題を繰り返し解くことが基本となる．線形計画問題は線形の等式と不等式制約の下，線形関数を最小化する問題であり，一般に次のように記述される．

$$\begin{aligned} \underset{x}{\text{Minimize}} \quad & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{subject to} \quad & A_1 \boldsymbol{x} = \boldsymbol{b}_1,\ A_2 \boldsymbol{x} \leq \boldsymbol{b}_2. \end{aligned} \tag{6}$$

ここで，A_1, A_2 は適当な次元の行列，$\boldsymbol{b}_1$, $\boldsymbol{b}_2$, $\boldsymbol{c}$ は適当な次元の定数ベクトル，$\boldsymbol{x}$ は変数ベクトルをそれぞれ表す．以下では，問題 (6) の形式で与えられる線形計画問題を数学的に取り扱いやすくするために，次のような標準形を定める．

$$\begin{aligned} \underset{x \in \mathbb{R}^n}{\text{Minimize}} \quad & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{subject to} \quad & A \boldsymbol{x} = \boldsymbol{b},\ \boldsymbol{x} \geq \boldsymbol{0}. \end{aligned} \tag{7}$$

ここで，$\boldsymbol{x} \in \mathbb{R}^n$ は変数ベクトル，$\boldsymbol{b} \in \mathbb{R}^n$, $\boldsymbol{c} \in \mathbb{R}^n$ は定数ベクトル，$A \in \mathbb{R}^{m \times n}$ は定数行列をそれぞれ表す．標準形で与えられる線形計画問題 (7) の特徴は，変数がすべて非負であることと，非負制約以外の条件が等式制約となっていることである．任意の線

形計画問題は，次のような操作によって問題 (7) で与えられる標準形に変換することができる．

- 最大化問題は目的関数を (-1) 倍することによって最小化問題へ変換する．
- 非負制約を持たない変数 (e.g., $\overline{\boldsymbol{x}}$) は，新しい非負変数 $\overline{\boldsymbol{x}}_1 \geq \mathbf{0}$, $\overline{\boldsymbol{x}}_2 \geq \mathbf{0}$ を導入して，$\overline{\boldsymbol{x}} := \overline{\boldsymbol{x}}_1 - \overline{\boldsymbol{x}}_2$ と置くことによって消去する．
- 不等号制約 $\boldsymbol{a}_i^\top \boldsymbol{x} \leq b_i$ は，新たな変数 z を導入し，等式制約 $\boldsymbol{a}_i^\top \boldsymbol{x} + z = b_i$ と非負制約 $z \geq 0$ に直す (不等号制約 $\boldsymbol{a}_i^\top \boldsymbol{x} \geq b_i$ についても同様).

問題 (7) を主問題とするとき，その双対問題は次のように記述される．

$$\begin{aligned} \underset{(y,z)\in\mathbb{R}^m\times\mathbb{R}^n}{\text{Maximize}} \quad & \boldsymbol{b}^\top \boldsymbol{y} \\ \text{subject to} \quad & A^\top \boldsymbol{y} + \boldsymbol{z} = \boldsymbol{c},\ \boldsymbol{z} \geq \mathbf{0}. \end{aligned} \tag{8}$$

主問題と双対問題の間には密接な関係があり，互いの目的関数値が最適解において一致することを述べた双対定理と呼ばれる重要な定理が成り立つ．

定理 2.1 [12, 定理 2.2] 主問題が最適解を持つならば，双対問題も最適解を持ち，そのときの双対問題の最適値は主問題の最適値に等しい．

定理 2.1 は，主問題または双対問題のいずれか一方が最適解を持つならば，もう一方の問題も最適解を持ち，その場合に両者の最適値は一致することを主張している．この定理より，次の系が得られる．

系 2.1 [12, 系 2.2] 主問題 (7) の実行可能解を $\boldsymbol{x}$，双対問題 (8) の実行可能解を $(\boldsymbol{y}, \boldsymbol{z})$ とする．このとき，次の条件 1 と条件 2 は同値である．

条件 1. $\boldsymbol{x}$ は主問題の最適解であり，$(\boldsymbol{y}, \boldsymbol{z})$ は双対問題の最適解である．

条件 2. $\boldsymbol{x}^\top \boldsymbol{z} = 0$ が成り立つ．

2.2 内点法

線形計画問題に対するアルゴリズムとして，内点法がよく知られている．内点法は，1984 年に N. Karmarkar によって提案された手法で，計算量が問題の規模の多項式オーダーで押さえられるため，大規模な問題に対しても高速に解が得られることが知られている．内点法には大きく分けて，主内点法と主双対内点法の 2 つがある．主内点法は，主問題または双対問題のいずれか一方の実行可能領域の内部に点列を生成する．一方，主双対内点法は，主問題と双対問題の双方の実行可能領域の内部に点列を生成する．

以下では，主問題 (7) と双対問題 (8) に対する最適解を求める主双対内点法について述べる．系 2.1 より，主問題 (7) と双対問題 (8) のそれぞれの実行可能解 $\boldsymbol{x}$ と $(\boldsymbol{y}, \boldsymbol{z})$

に対して，

$$
\begin{cases}
A\boldsymbol{x} = \boldsymbol{b} \\
A^\top \boldsymbol{y} + \boldsymbol{z} = \boldsymbol{c} \\
x_i z_i = 0 \quad (i = 1, 2, \ldots, n) \\
\boldsymbol{x} \geq \boldsymbol{0},\ \boldsymbol{z} \geq \boldsymbol{0}
\end{cases} \tag{9}
$$

が成り立つならば，$\boldsymbol{x}$ は主問題 (7) の最適解であり，$(\boldsymbol{y}, \boldsymbol{z})$ は双対問題 (8) の最適解である．したがって，式 (9) を満たすようなベクトルの組 $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ を求めることによって，主問題と双対問題を同時に解くことができる．以下では，式 (9) を満たすベクトルの組 $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ を求める問題を主双対問題と呼ぶことにする．

主双対アフィンスケーリング法は主双対問題 (9) を解く手法の一つであり，Monterio, Adler, Resende[7] によって提案された．この手法では，双対ギャップ $(\boldsymbol{x}^k)^\top \boldsymbol{z}^k$ の値が単調減少するように実行可能内点列 $\left\{(\boldsymbol{x}^k, \boldsymbol{y}^k, \boldsymbol{z}^k) \mid k = 0, 1, 2, \ldots\right\}$ を生成することを考える．ここで，主双対問題 (9) に対する実行可能内点の集合は次のように定義される．

定義 2.1　ベクトルの組 $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ が集合

$$
\mathcal{F}_{PD} := \left\{(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) \mid A\boldsymbol{x} = \boldsymbol{b},\ A^\top \boldsymbol{y} + \boldsymbol{z} = \boldsymbol{c},\ \boldsymbol{x} > \boldsymbol{0},\ \boldsymbol{z} > \boldsymbol{0}\right\}
$$

の要素であるとき，$(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ は主双対問題 (9) の実行可能内点であるという．

主双対アフィンスケーリング法の具体的なアルゴリズムは次のように記述される．

主双対アフィンスケーリング法

Step 0.　十分小さな定数 $\varepsilon > 0$ を定める．$k := 0$ とし，初期点を $(\boldsymbol{x}^0, \boldsymbol{y}^0, \boldsymbol{z}^0) \in \mathcal{F}_{PD}$ とする．

Step 1.　収束条件 $(\boldsymbol{x}^k)^\top \boldsymbol{z}^k \leq \varepsilon$ が成り立つならば計算終了する．さもなければ，線形方程式系

$$
\begin{pmatrix} A & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & A^\top & I \\ Z_k & \boldsymbol{0} & X_k \end{pmatrix}
\begin{pmatrix} \Delta \boldsymbol{x}^k \\ \Delta \boldsymbol{y}^k \\ \Delta \boldsymbol{z}^k \end{pmatrix}
= -\begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ X_k \boldsymbol{z}^k \end{pmatrix} \tag{10}
$$

を解き，$(\Delta \boldsymbol{x}^k, \Delta \boldsymbol{y}^k, \Delta \boldsymbol{z}^k)$ を求める．

Step 2.　条件 $(\boldsymbol{x}^k + \alpha_P^k \Delta \boldsymbol{x}^k, \boldsymbol{z}^k + \alpha_D^k \Delta \boldsymbol{z}^k) > 0$ を満たすようなパラメータ α_P^k と α_D^k の値を定め，点 $(\boldsymbol{x}^{k+1}, \boldsymbol{y}^{k+1}, \boldsymbol{z}^{k+1}) \in \mathcal{F}_{PD}$ を次式で定める．

$$
\begin{pmatrix} \boldsymbol{x}^{k+1} \\ \boldsymbol{y}^{k+1} \\ \boldsymbol{z}^{k+1} \end{pmatrix}
= \begin{pmatrix} \boldsymbol{x}^k + \alpha_p^k \Delta \boldsymbol{x}^k \\ \boldsymbol{y}^k + \alpha_D^k \Delta \boldsymbol{y}^k \\ \boldsymbol{z}^k + \alpha_D^k \Delta \boldsymbol{z}^k \end{pmatrix}. \tag{11}
$$

Step 3.　$k := k + 1$ とし，*Step 1* へ戻る．

ここで式 (10) における $X_k = \mathrm{diag}(\boldsymbol{x}^k)$ は $\boldsymbol{x}^k$ の各要素を対角成分とする対角行列を，$Z_k = \mathrm{diag}(\boldsymbol{z}^k)$ は $\boldsymbol{z}^k$ の各要素を対角成分とする対角行列をそれぞれ表す．

主双対アフィンスケーリング法のアルゴリズムの *Step 2* において，主問題と双対問題のステップサイズが等しい場合，次のような定理が成り立つことが知られている．

定理 2.2 [12, 定理 4.1] 主双対アフィンスケーリング法のアルゴリズムの *Step 2* において，条件 $(\boldsymbol{x}^k + \alpha_P^k \Delta\boldsymbol{x}^k, \boldsymbol{z}^k + \alpha_D^k \Delta\boldsymbol{z}^k) > 0$ を満たすような，あるパラメータ $\alpha > 0$ が存在して，$\alpha_P^k = \alpha$, $\alpha_D^k = \alpha$ とする．このとき，

$$(\boldsymbol{x}^{k+1})^\top \boldsymbol{z}^{k+1} = (1-\alpha)(\boldsymbol{x}^k)^\top \boldsymbol{z}^k$$

が成り立つ．

定理 2.2 により，任意の $\alpha > 0$ に対して $(\boldsymbol{x}^{k+1})^\top \boldsymbol{z}^{k+1} > (\boldsymbol{x}^k)^\top \boldsymbol{z}^k$ が成り立つので，主双対アフィンスケーリング法で生成される点列の双対ギャップは単調に減少する．

3 LSIP に対するアルゴリズム

3.1 緩和切除平面法

まず，Wu, Fang, Lin [11] によって提案された緩和切除平面法 (RCP 法) について述べる．RCP 法では，アルゴリズムの第 k 回目の反復において，集合 T の部分集合 T_k で特徴づけられる部分問題 $\mathrm{LP}(T_k)$ を厳密に解き，暫定解 $\boldsymbol{x}^k$ を求める．暫定解 $\boldsymbol{x}^k$ が LSIP(1) の解であるためには，すべての $\boldsymbol{t} \in T$ に対して $\boldsymbol{a}(\boldsymbol{t})^\top \boldsymbol{x}^k - b(\boldsymbol{t}) \geq 0$ を満たしている必要がある．もし，暫定解 $\boldsymbol{x}^k$ において $\boldsymbol{a}(\boldsymbol{t})^\top \boldsymbol{x}^k - b(\boldsymbol{t}) < 0$ なる添字 $\boldsymbol{t}_{k+1}$ が存在するならば，$T_{k+1} := T_k \cup \{\boldsymbol{t}_{k+1}\}$ として，次の反復で解く部分問題 $\mathrm{LP}(T_{k+1})$ を構成し，反復を繰り返す．RCP 法は具体的に次のように記述される．

Alg 1: RCP 法

Step 0. 十分小さな定数 $\delta > 0$ を定める．集合 T の中から有限個の要素を任意に選び $T_0 := \{\boldsymbol{t}_1, \boldsymbol{t}_2, \ldots, \boldsymbol{t}_m\} \subset T$ とする．集合 T_0 によって特徴づけられる部分問題 $\mathrm{LP}(T_0)$ を解き，最適解 $\boldsymbol{x}^0$ を求める[*5]．$k := 0$ とする．

Step 1. $\boldsymbol{a}(\boldsymbol{t})^\top \boldsymbol{x}^k - b(\boldsymbol{t}) < -\delta$ を満たすような添字 $\boldsymbol{t}_{\mathrm{new}}^k \in T \setminus T_k$ を見つける．もし，$\boldsymbol{a}(\boldsymbol{t})^\top \boldsymbol{x}^k - b(\boldsymbol{t}) < -\delta$ を満たす $\boldsymbol{t}_{\mathrm{new}}^k$ が存在しない[*6] ならば計算終了．さもなければ $T_{k+1} := T_k \cup \{t_{\mathrm{new}}^k\}$ とする．

Step 2. 集合 T_{k+1} で特徴づけられる部分問題 $\mathrm{LP}(T_{k+1})$ を解き，最適解 $\boldsymbol{x}^{k+1}$ を求める．$k := k+1$ とおき，*Step 1* へ戻る．

---

*5 $\mathrm{LP}(T_0)$ が最適解を持つように，T_0 として十分な数の添字を選ぶ．

*6 すべての $\boldsymbol{t} \in T$ に対して $\boldsymbol{a}(\boldsymbol{t})^\top \boldsymbol{x}^k - b(\boldsymbol{t}) \geq -\delta$ であることを意味する．

Step 1 では，従来の切除平面法で解かなければならなかった下位問題 (5) を解く代わりに，$\boldsymbol{a}(\boldsymbol{t})^\top \boldsymbol{x}^k - b(\boldsymbol{t}) < -\delta$ を満たすような添字 $\boldsymbol{t}^k_{\mathrm{new}} \in T \setminus T_k$ を見つけることによって添字集合の更新を行なっている．Wu *et al.* は RCP 法を適用することによって，従来の切除平面法よりも高速に解が得られることを数値実験で確認している．

3.2　提案手法

次に，本稿で提案する切除平面法を高速化するアプローチについて述べる．提案する手法では，RCP 法の *Step 0* と *Step 2* において厳密に解くことが前提となっている LP を非厳密に解くことにより全体の高速化を図るアプローチを導入する．ここで部分問題は (4) の形式で与えられる線形計画問題であるが，以下では (7) の形式で与えられる問題を部分問題として考える．なお．問題 (4) から問題 (7) へ再定式化できることは 2 節で確認した通りである．さて，提案手法では部分問題の厳密な最適解を求める代わりに，次のように定義される精度 $\beta(\geq 0)$ の近似解を求めることを考える．

定義 3.2　線形計画問題 (7) とその双対問題 (8) に対して

$$\max\left\{\frac{\|A\boldsymbol{x}-\boldsymbol{b}\|_2}{\max(1,\|\boldsymbol{b}\|_2)}, \frac{\|A^\top\boldsymbol{y}+\boldsymbol{z}-\boldsymbol{c}\|_2}{\max(1,\|\boldsymbol{c}\|_2)}, \frac{|\boldsymbol{c}^\top\boldsymbol{x}-\boldsymbol{b}^\top\boldsymbol{y}|}{\max(1,|\boldsymbol{c}^\top\boldsymbol{x}|,|\boldsymbol{b}^\top\boldsymbol{y}|)}\right\} \leq \beta \tag{12}$$

を満たすようなベクトル $\boldsymbol{x}$ および $(\boldsymbol{y},\boldsymbol{z})$ を精度 $\beta(\geq 0)$ の近似解とよぶ．

定義 3.2 において，$A\boldsymbol{x}-\boldsymbol{b}$ は主問題 (7) の残差，$A^\top\boldsymbol{y}+\boldsymbol{z}-\boldsymbol{c}$ は双対問題 (8) の残差，$\boldsymbol{c}^\top\boldsymbol{x}-\boldsymbol{b}^\top\boldsymbol{y}$ は双対ギャップをそれぞれ表し，$\|\cdot\|_2$ は本稿を通してユークリッドノルムを表すことにする．また，式 (12) における各分母は誤差の相対化をそれぞれ行っている．$\beta=0$ において式 (12) を満たすようなベクトル $\boldsymbol{x}$ および $(\boldsymbol{y},\boldsymbol{z})$ は，主問題 (7) と双対問題 (8) の厳密な最適解にそれぞれ対応し，MATLAB Optimization Toolbox の “`linprog`” では，β を十分小さな定数 (e.g., $\beta=10^{-6}$) とすることによって LP の最適解を求めている．本研究で提案する手法では，部分問題として解くべき LP を定義 3.2 で定められる精度 $\beta(\geq 0)$ の近似解で留めることを考える．具体的なアルゴリズムは次のように記述される．

ALG 2: RCP 法 (高速化)

Step 0.　十分小さな定数 $\gamma > 0$ と適当な正のパラメータ $M > 0,\ \rho > 0$ をそれぞれ定める．次の条件 (i), (ii) を満たすような数列 $\{\beta_k\}_{k=0}^{\infty}$ を定める．

$$\text{(i)}\quad \beta_k \geq 0 \quad (k=0,1,2,\ldots) \qquad \text{(ii)}\quad \lim_{k\to\infty}\beta_k = 0$$

集合 T の中から有限個の要素を任意に選び $T_0 := \{\boldsymbol{t}_1,\boldsymbol{t}_2,\ldots,\boldsymbol{t}_m\} \subset T$ とおく．集合 T_0 によって特徴づけられる部分問題 LP(T_0) に対する精度 β_0 の近似解 $\tilde{\boldsymbol{x}}^0$ を求める．$k := 0$ とする．

Step 1.　次の不等式を満たす $\boldsymbol{t}^k_{\mathrm{new}} \in T \setminus T_k$ を見つける．

$$\boldsymbol{a}(\boldsymbol{t})^\top \tilde{\boldsymbol{x}}^k - b(\boldsymbol{t}) < -M\rho\beta_k. \tag{13}$$

$T_{k+1} := T_k \cup \{\boldsymbol{t}^k_{\mathrm{new}}\}$ とおき，*Step 3* へ進む．もし，式 (13) を満たす $\boldsymbol{t}^k_{\mathrm{new}} \in T \setminus T_k$ が存在しない[*7] ならば *Step 2* へ進む．

Step 2.　$M\rho\beta_k \leq \gamma$ であるならば計算終了．さもなければ，$\boldsymbol{t}^k_{\mathrm{new}} := \underset{t \in T}{\operatorname{argmin}}\{\boldsymbol{a}(\boldsymbol{t})^\top \tilde{\boldsymbol{x}}^k - b(\boldsymbol{t})\}$ を計算し，$T_{k+1} := T_k \cup \{\boldsymbol{t}^k_{\mathrm{new}}\}$ とする．*Step 3* へ進む．

Step 3.　集合 T_{k+1} によって特徴づけられる部分問題 $\mathrm{LP}(T_{k+1})$ に対する精度 β_{k+1} の $\tilde{\boldsymbol{x}}^{k+1}$ を求める．$k := k+1$ とし，*Step 1* へ戻る．

Step 0 で設定する十分小さな定数 $\gamma > 0$ は RCP 法における $\delta > 0$ に相当し，解くべき LSIP の解に対する精度を表す．また，数列 $\{\beta_k\}_{k=0}^{\infty}$ は，$k \to \infty$ の極限で 0 に収束する非負数列であり，$\beta_k := 2^{-k}\,(k = 0, 1, 2 \ldots)$ などが例として挙げられる．この数列 $\{\beta_k\}_{k=0}^{\infty}$ は，定義 3.2 における精度 $\beta \geq 0$ に相当し，提案手法では反復を重ねる度に部分問題 $\mathrm{LP}(T_k)$ に対する解への精度が厳しくなることを要求している．

Step 1 では，添字集合 T_k を更新するためのステップである．既存の RCP 法では，$\boldsymbol{a}(\boldsymbol{t})^\top \boldsymbol{x}^k - b(\boldsymbol{t}) < -\delta$ を解くことにより，新たな制約条件として加える添字 $\boldsymbol{t}^k_{\mathrm{new}} \in T \setminus T_k$ を見つけることができた．しかし，提案アプローチでは $\mathrm{LP}(T_k)$ に対して近似最適解を求めるに留めているため，同じ手法を直接適用することはできない．そこで，添字集合 T_k を更新するために，次の定理を考える．

定理 3.3　部分問題 $\mathrm{LP}(T_k)$ の最適解を $\boldsymbol{x}^k$，精度 $\beta_k \geq 0$ の近似解を $\tilde{\boldsymbol{x}}^k = \tilde{\boldsymbol{x}}^k(\beta_k)$ とする．さらに任意の $\beta \geq 0$ に対して，$\|\boldsymbol{x}^k - \tilde{\boldsymbol{x}}^k\|_2 \leq \rho\beta_k$ を満たすような正の実数 $\rho > 0$ が存在すると仮定する．このとき，$\boldsymbol{a}(\boldsymbol{t})^\top \tilde{\boldsymbol{x}}^k - b(\boldsymbol{t}) < -M\rho\beta_k$ を満たすような任意の $\boldsymbol{t} \in T$ に対して，$\boldsymbol{a}(\boldsymbol{t})^\top \boldsymbol{x}^k - b(\boldsymbol{t}) < 0$ が成り立つ．ただし，$M := \sup_{t \in T} \|\boldsymbol{a}(\boldsymbol{t})\|_2$ である．

証明.　式 (13) を満たす任意の $\boldsymbol{t} \in T \setminus T_k$ に対して

$$\begin{aligned}\boldsymbol{a}(\boldsymbol{t})^\top \tilde{\boldsymbol{x}}^k - b(\boldsymbol{t}) &= \boldsymbol{a}(\boldsymbol{t})^\top \boldsymbol{x}^k - b(\boldsymbol{t}) + \boldsymbol{a}(\boldsymbol{t})^\top (\tilde{\boldsymbol{x}}^k - \boldsymbol{x}^k) \\ &< -M\rho\beta_k + \|\boldsymbol{a}(\boldsymbol{t})\|_2 \cdot \|\tilde{\boldsymbol{x}}^k - \boldsymbol{x}^k\|_2 \\ &\leq -M\rho\beta_k + M\rho\beta_k \\ &= 0\end{aligned}$$

が成り立つ．ここで 2 行目の不等式は Cauchy-Schwarz の不等式より，3 行目の不等式は本定理の仮定より成り立つ．■

定理 3.3 より，$\|\boldsymbol{x}^k - \tilde{\boldsymbol{x}}^k\|_2 \leq \rho\beta_k$ を満たすような $\rho > 0$ の値を事前に予測することができれば，$\mathrm{LP}(T_k)$ の厳密な最適解 $\boldsymbol{x}^k$ を実際に計算しなくても，近似最適解 $\tilde{\boldsymbol{x}}^k$ の情報だけで，$\boldsymbol{a}(\boldsymbol{t})^\top \boldsymbol{x}^k - b(\boldsymbol{t}) < 0$ を満たす添字 $\boldsymbol{t} \in T \setminus T_k$ を見つけることができる．

[*7] すべての $\boldsymbol{t} \in T$ に対して $\boldsymbol{a}(\boldsymbol{t})^\top \tilde{\boldsymbol{x}}^k - b(\boldsymbol{t}) \geq -M\rho\beta_k$ であることを意味する．

Step 2 は，*Step 1* で式 (13) を満たすような添字 $\boldsymbol{t}_{\text{new}}^k \in T \setminus T_k$ が存在しないときのステップである．まず，$M\rho\beta_k$ の値が *Step 0* で定めた γ 以下である場合は暫定解 $\tilde{\boldsymbol{x}}^k$ を解として出力し，計算を終了する．一方，$M\rho\beta_k > \gamma$ であるならば，$\boldsymbol{t}_{\text{new}}^k := \operatorname*{argmin}_{t \in T}\{\boldsymbol{a}(\boldsymbol{t})^\top \tilde{\boldsymbol{x}}^k - b(\boldsymbol{t})\}$ を計算し，$T_{k+1} := T_k \cup \{\boldsymbol{t}_{\text{new}}^k\}$ とすることで更新する．

上記の手法 (Alg 2) で $\beta_k := 0\,(k = 0, 1, 2 \ldots)$ とすると，RCP 法において $\delta := 0$ としたときに対応する．このとき，Alg 2 における *Step 2* の $M\rho\beta_k \le \gamma$ は $0 \le \gamma$ となるため，常に成り立つ．したがって，Alg 2 の *Step 1* において，式 (13) を満たすような添字 $\boldsymbol{t}_{\text{new}}^k \in T \setminus T_k$ が存在しなければ計算が終了されることを意味しており，これは Alg 1 の *Step 1* に対応している．

3.3　収束解析

本項では，提案手法が $\gamma \to 0$ の極限において，Alg 2 によって得られる出力が LSIP(1) の最適解に収束することを示す．はじめに，定義 3.2 で記述される精度 $\beta(\ge 0)$ の近似解に対して次のエラーバウンド性が成り立つことを仮定する．

仮定 A　部分問題 LP(T_k) は唯一の最適解 $\boldsymbol{x}^k$ をもつとする．さらに，それに対する任意の精度 β_k の近似解を $\tilde{\boldsymbol{x}}^k$ としたとき，$\|\boldsymbol{x}^k - \tilde{\boldsymbol{x}}^k\|_2 \le \rho\beta_k$ を満たす k に依らない正数 $\rho > 0$ が存在する．

仮定 A のもと，次の定理が成り立つ．

定理 3.4　仮定 A が成り立つとする．また，所与の $\gamma > 0$ に対して，Alg 2 は有限の反復回数 $k_0 = k_0(\gamma)$ で終了するとする．このとき，LSIP(1) の最適解集合 $\mathcal{S}$ に対して $\lim_{\gamma\to 0} \operatorname{dist}(\tilde{\boldsymbol{x}}^{k_0}, \mathcal{S}) = 0$ が成り立つ．

証明.　まず，LP(T_{k_0}) の厳密な最適解 $\boldsymbol{x}^{k_0}$ に対して，以下の不等式が成り立つ．

$$
\begin{aligned}
\min_{t\in T}\left[\boldsymbol{a}(\boldsymbol{t})^\top \boldsymbol{x}^{k_0} - b(\boldsymbol{t})\right] &= \min_{t\in T}\left[\boldsymbol{a}(\boldsymbol{t})^\top \tilde{\boldsymbol{x}}^{k_0} - b(\boldsymbol{t}) + \boldsymbol{a}(\boldsymbol{t})^\top(\boldsymbol{x}^{k_0} - \tilde{\boldsymbol{x}}^{k_0})\right] \\
&\ge \min_{t\in T}\left[\boldsymbol{a}(\boldsymbol{t})^\top \tilde{\boldsymbol{x}}^{k_0} - b(\boldsymbol{t})\right] + \min_{t\in T}\left[\boldsymbol{a}(\boldsymbol{t})^\top(\boldsymbol{x}^{k_0} - \tilde{\boldsymbol{x}}^{k_0})\right] \\
&\ge \min_{t\in T}\left[\boldsymbol{a}(\boldsymbol{t})^\top \tilde{\boldsymbol{x}}^{k_0} - b(\boldsymbol{t})\right] + \min_{t\in T}\left(-\|\boldsymbol{a}(\boldsymbol{t})\|_2 \cdot \|\tilde{\boldsymbol{x}}^{k_0} - \tilde{\boldsymbol{x}}^{k_0}\|_2\right) \\
&\ge -M\rho\beta_{k_0} - M\rho\beta_{k_0} \\
&\ge -2\gamma
\end{aligned}
$$

ここで，最初の不等式は min の性質より，2 つめの不等式は Cauchy-Schwarz の不等式より，3 つめの不等式は式 (13) を満たす $\boldsymbol{t} \in T$ が存在しないことと，仮定 A および $M = \sup_{t\in T}\|\boldsymbol{a}(\boldsymbol{t})\|_2$ より，4 つめの不等式は Alg 2 の終了条件より成り立つ．

一方，LSIP(1) の実行可能領域は LP(T_{k_0}) の実行可能領域に含まれるので，LSIP(1) の最適解を $\boldsymbol{x}^*$ に対して不等式 $\boldsymbol{c}^\top\boldsymbol{x}^* \ge \boldsymbol{c}^\top\boldsymbol{x}^{k_0}$ が成り立つ．これより，

$\lim_{\gamma \to 0} \mathrm{dist}(\boldsymbol{x}^{k_0}, \mathcal{S}) = 0$ を得る [4].

さらに，$\|\boldsymbol{x}^{k_0} - \tilde{\boldsymbol{x}}^{k_0}\|_2 \leq \rho\beta_{k_0} \leq M^{-1}\gamma$ であることに注意すると，

$$\mathrm{dist}(\tilde{\boldsymbol{x}}^{k_0}, \mathcal{S}) \leq \mathrm{dist}(\boldsymbol{x}^{k_0}, \mathcal{S}) + \|\boldsymbol{x}^{k_0} - \tilde{\boldsymbol{x}}^{k_0}\|_2 \leq \mathrm{dist}(\boldsymbol{x}^{k_0}, \mathcal{S}) + M^{-1}\gamma$$

であるので，$\gamma \to 0$ の極限をとれば，$\lim_{\gamma \to 0} \mathrm{dist}(\tilde{\boldsymbol{x}}^{k_0}, \mathcal{S}) = 0$ を得る． ■

4　数値実験

本節では，RCP 法 (Alg 1) と提案手法 (Alg 2) の比較を行う．実験には 1 節で述べたチェビシェフ近似問題 (3) において，$\varphi(t) = -\sqrt{2\pi t - t^2}$，$P(t, \boldsymbol{x}) := \sum_{i=1}^{n} \cos\{(i-1)t\}x_i$，$[\alpha, \beta] := [0, 2\pi]$ としたときの問題を扱う．この問題は次のように記述される．

$$\begin{aligned} &\underset{(x,\eta)\in\mathbb{R}^{n+1}}{\text{Minimize}} && \eta \\ &\text{subject to} && -\eta \leq \sum_{i=1}^{n} x_i \cos(i-1)t - (-\sqrt{2\pi t - t^2}) \leq \eta \quad (\forall t \in [0, 2\pi]). \end{aligned} \tag{14}$$

この実験では，RCP 法の *Step 0* における定数 δ の値を $\delta = 10^{-6}$ と設定した．また，提案手法におけるパラメータ M の値は，$M = \sqrt{n}$ と設定し，数列 $\{\beta_k\}_{k=0}^{\infty}$ は $\beta_k = 10^{-3} \cdot (\sqrt{2})^{-k}$ $(k = 0, 1, 2, \ldots)$ と設定した．さらに，*Step 0* における集合 T_0 はどちらの手法でも $T_0 := \widetilde{T}_0 \cup \{0, 2\pi\}$ で構成した．ここで，集合 $\widetilde{T}_0$ は n 個の要素を含む集合で，各要素は 0 以上 2π 以下の一様乱数で構成される．

表 1 は，それぞれの手法を 20 回ずつ試行した結果の平均値を表し，n は決定変数 $\boldsymbol{x}$ の次元，♯ite(out) は RCP 法の反復回数 (外部反復回数)，♯ite(in) は部分問題である LP を解くためにかかった反復回数 (内部反復回数)，cpu(s) は計算時間 (秒) をそれぞれ表す．表 1 より次のようなことが考察できる．まず計算時間に関しては，決定変数 $\boldsymbol{x}$ の次元が低いときは，RCP 法と高速化を施した提案手法のいずれの場合で解いても計算時間に大きな差は見られない．一方，決定変数 $\boldsymbol{x}$ の次元が高くなるにつれて，提案手法の方が計算時間が短縮されていることが観察できる．また反復回数に関しては，外部反復回数 ♯ite(out) は両者でほとんど相違がなかったが，内部反復回数 ♯ite(in) は提案手法の方が少ないことが確認できる．この結果から，部分問題を非厳密に解くことによって内部反復回数が抑えられることが観察できる．

表 1　RCP 法と提案手法の比較結果 (20 回の平均値)

n	RCP			RCP(高速化)		
	♯ite		cpu	♯ite		cpu
	(out)	(in)	(s)	(out)	(in)	(s)
25	26.8	7.2	0.28	28.1	6.2	0.31
50	47.6	8.2	1.05	48.2	7.6	1.06
100	73.6	9.0	7.20	74.7	8.3	7.02
150	85.5	8.7	21.38	84.9	7.7	19.32
200	100.7	9.0	54.59	98.2	7.1	44.46
250	105.4	9.5	140.53	104.9	6.6	105.44

5　結論

本稿では，線形半無限計画問題に対する既存アルゴリズムである RCP 法において，厳密に解くことが前提となっている部分問題を非厳密に解くアプローチを提案し，その収束解析を行なった．また，数値実験を通して，決定変数が高次元の場合において RCP 法よりも提案手法の方がより高速に解が得られることを確認した．

今後の課題として，部分問題の近似最適解に対する理論的な評価を与えることが挙げられる．特に本稿では，定理 3.3 で仮定されている定数 $\rho > 0$ に関して十分に議論できておらず，その値が提案手法にどのような影響を及ぼすのか明らかになっていない．

謝辞

This work was supported by the Research Institute for Mathematical Sciences, an International Joint Usage/Research Center located in Kyoto University.

参考文献

[1] B. Betrò, *An accelerated central cutting plane algorithm for linear semi-infinite programming*, Mathematical Programming, 101 (2004), pp. 479–495.

[2] M. A. Goberna, *Linear semi-infinite optimization: recent advances*, in Continuous Optimization, Springer, 2005, pp. 3–22.

[3] M. A. Goberna and M. A. López, *Linear semi-infinite optimization*,

vol. 2, Wiley, 1998.

[4] S. Hayashi and S.-Y. Wu, *An explicit exchange algorithm for linear semi-infinite programming problems with second-order cone constraints*, SIAM Journal on optimization, 20 (2009), pp. 1527–1546.

[5] R. Hettich and K. O. Kortanek, *Semi-infinite programming: theory, methods, and applications*, SIAM review, 35 (1993), pp. 380–429.

[6] H. Lu, G. Huang, and L. He, *A semi-infinite analysis-based inexact two-stage stochastic fuzzy linear programming approach for water resources management*, Engineering Optimization, 41 (2009), pp. 73–85.

[7] R. D. Monteiro and I. Adler, *Interior path following primal-dual algorithms. part I: Linear programming*, Mathematical programming, 44 (1989), pp. 27–41.

[8] G. Still, *Discretization in semi-infinite programming: the rate of convergence*, Mathematical programming, 91 (2001), pp. 53–69.

[9] K. L. Teo, X. Yang, and L. S. Jennings, *Computational discretization algorithms for functional inequality constrained optimization*, Annals of Operations Research, 98 (2000), pp. 215–234.

[10] S.-P. Wu, S. Boyd, and L. Vandenberghe, *Fir filter design via spectral factorization and convex optimization*, in Applied and Computational Control, Signals, and Circuits, Springer, 1999, pp. 215–245.

[11] S.-Y. Wu, S. Fang, and C.-J. Lin, *Relaxed cutting plane method for solving linear semi-infinite programming problems*, Journal of Optimization Theory and Applications, 99 (1998), pp. 759–779.

[12] 小島政和, 土谷隆, 水野眞治, 矢部博, 内点法, 朝倉書店, 2001.